$\boxed{求\ dp[n-1][k][0]}$

dp[i][k][0 or 1]

i表天数 $0 \le i \le n-1$

k表最多交易数 $1 \le k \le K$

s: [0,1] 表当前持有股票状态（1有 0无）

$\Rightarrow$

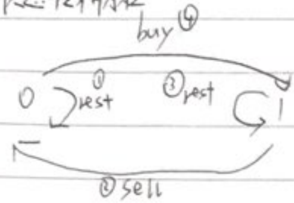
for $0 \le i \le n-1$

  for $1 \le k \le K$

    for s = 0, 1

      dp[i][k][s] = max(buy, sell, rest)

状态转移方程



dp[i][k][0] = max(dp[i-1][k][0] ①, dp[i-1][k][1] + prices[i] ②)

dp[i][k][1] = max(dp[i-1][k][1] ③, dp[i-1][k-1][0] - prices[i] ④)

注意: k-1 因为要花一次交易去buy

基础边界 dp[-1][k][0] = 0 未开始时利润为0

$-\infty$ 表不可能条件 dp[-1][k][1] = $-\infty$ 未开始时不可能有股票 ∴ $-\infty$

dp[i][0][0] = 0 不让买卖 利润为0

dp[i][0][1] = $-\infty$ 不让买卖 不可能有股票 ∴ $-\infty$

1. 对于k=1 dp[i][1][0] = max(dp[i-1][1][0], dp[i-1][1][1] + prices[i])

dp[i][1][1] = max(dp[i-1][1][1], dp[i-1][0][0] - prices[i])

// 压缩(k为1不变) 成二维数组　　不让买卖所以为0，所以为 -prices[i]

dp[i][0] = max(dp[i-1][0], dp[i-1][1] + prices[i])　边界弄好 $\Rightarrow$ return dp[n-1][0]

dp[i][1] = dp[i-1][1]

2. 对于 $k=+\infty$ ~~dp[i][k]~~ $k=+\infty$时可以认为 k与k-1是一样的

∴ dp[i][k][0] = max(dp[i-1][k][0], dp[i-1][k][1] + prices[i])

dp[i][k][1] = max(dp[i-1][k][1], dp[i-1][k][0] - prices[i])

// 压缩 k=k不变，成二维数组

dp[i][0] = max(dp[i-1][0], dp[i-1][1] + prices[i])

dp[i][1] = max(dp[i-1][1], dp[i-1][0] - prices[i])　边界弄好 $\Rightarrow$ return dp[n-1][0]

3. 对于 k=2 dp[i][k][0] = max(dp[i-1][k][0], dp[i-1][k][1] + prices[i])

dp[i][k][1] = max(dp[i-1][k][1], dp[i-1][k-1][0] - prices[i])